

Installation de l'environnement de développement

Contexte

La société **XILIM**, Entreprise de **S**ervices du **N**umérique (anciennement **S**ociété de **S**ervices en **I**ngénierie **I**nformatique), souhaite se doter d'un nouvel outil de *monitoring* réseau. Le développement de cette application se fera en interne. Actuellement, le projet est à l'étude. Seuls le système d'exploitation Linux (distribution Debian 9 ou 10) et le SGBDR (Système de Gestion de Bases de Données Relationnelles) *MariaDB* ont été définis. Cependant, le choix de l'architecture logicielle et des technologies exploitées (Apache/PHP ou Werkzeug/Python/Flask) n'est pas encore arrêté.

Stagiaire dans la société XILIM, vous intégrez l'équipe en charge de la phase préparatoire du projet.

Cette mission fait l'objet d'un rapport au format PDF (AP1-Mission-4-NOM-Prénom.pdf) qui fera apparaître :

- Le contexte ;
- L'intitulé de la mission ;
- Le nom et le prénom ;
- Les commandes saisies, les résultats obtenus, des explications...
- Les difficultés rencontrées.

Le rapport est à envoyer par courrier électronique (Objet : [AP1][4] NOM Prénom).

1- Préambule

1. Créer la machine virtuelle VM-XILIM-Monitoring en clonant (VM > Manage > Clone...) la machine virtuelle VM-DEV-WEB créée dans la mission 2 du module AP1.
2. [devadmin] Créer le dossier ~/xilim/projet/bd/.

2- Base de données

Une base de données dédiée aux tests, nommée *bdtests* sera utilisée au cours de la phase d'étude.

3. [devadmin] Écrire le script (requêtes en langage SQL) ~/xilim/projet/bd/bdtests.sql dont le code source est donné ci-dessous (ce script a pour rôle de créer la base de données de tests et d'y enregistrer une liste de postes de travail dans une table nommée Poste) :

```
drop database if exists bdtests ;
create database if not exists bdtests ;
use bdtests ;
create table Poste (
    id int(11) not null auto_increment ,
    nom varchar(20) not null ,
    salle varchar(5) default null ,
    primary key(id)
) ;

insert into Poste(nom,salle) values('A104-P1','a104') ;
insert into Poste(nom,salle) values('A207b-P2','a207b') ;
insert into Poste(nom,salle) values('A202-P1','a202') ;
insert into Poste(nom,salle) values('A104-P10','a104') ;
insert into Poste(nom,salle) values('A104-P5',NULL) ;
insert into Poste(nom,salle) values('A207A-P7','a207a') ;
```

4. [devadmin] Ouvrir une session root.

```
| su
```

5. [root] Exécuter le script `bdtests.sql` dans le but de créer et de peupler la base de données `bdtests` :
| `mariadb < bdtests.sql`
6. [root] Exécuter le client *MariaDB* en mode interactif :
| `mariadb`
7. Afficher la liste des bases de données hébergées et vérifier la présence de la base de données `bdtests` :
| `show databases ;`
8. Sélectionner la base de données `bdtests` (la base de données `bdtests` devient la base courante) :
| `use bdtests ;`
9. Afficher la liste des tables et vérifier la présence de la table `Poste` :
| `show tables ;`
10. Afficher la structure de la table `Poste` :
| `desc Poste ;`
11. Afficher les données présentes dans la table `Poste` :
| `select * from Poste ;`
12. Le modèle relationnel de la base de données `bdtests` est le suivant :
`Poste(id , nom , salle)`
Comparer la requête SQL de création (*question 3*), la structure (*question 10*), le modèle relationnel (*question 12*) et le contenu (*question 11*) de la table `Poste`. Que peut-on en déduire ?
13. Créer le compte local *MariaDB* superviseur qui a pour mot de passe `azerty` (se documenter sur la requête SQL de création de compte `create user`).
14. Vérifier l'enregistrement du compte superviseur dans la base `mysql` (table `user`).
Information : Le nombre de colonnes de cette table est conséquent. Pour une lecture plus aisée du contenu de la table, il est recommandé d'activer le mode d'affichage vertical (`select ... \G`).
15. Donner tous les droits (consultation, insertion...) sauf celui d'accorder des droits, sur la base de données `bdtests` au compte superviseur (se documenter sur la requête SQL d'attribution de droits `grant`).
| `grant ... ;`
| `flush privileges ;`
16. Vérifier les droits attribués au compte superviseur (`show grants`).
17. Vérifier l'enregistrement des droits accordés au compte superviseur dans la base `mysql` (table `db`).
18. Dédire des manipulations précédentes le rôle de la base de données `mysql`.
19. Quitter le client *MariaDB* en mode interactif (`exit`) puis mettre fin à la session `root` (`exit`).
20. [devadmin] Se connecter au SGBDR avec le compte local *MariaDB* superviseur.
| `mariadb -h localhost -u superviseur -p`
21. Réitérer les étapes 7 à 11, puis quitter le client *MariaDB* en mode interactif.

3- Environnement PHP

Des scripts PHP minimalistes écrits en langage PHP seront implémentés pour réaliser les tests de bon fonctionnement des différents composants logiciels exploités dans le projet (serveur HTTP Apache 2, interpréteur PHP et connecteur PHP-MariaDB).

22. [devadmin] Accéder à l'URL `http://127.0.0.1/` dans un navigateur *web*.
23. [root] Donner tous les droits d'accès au dossier `/var/www/html` à tous les utilisateurs du système.
24. [devadmin] Créer le dossier `/var/www/html/webtests`.
25. [devadmin] Créer le fichier `accueil.html`, dont le code est donné ci-dessous, dans le dossier `/var/www/html/webtests`, puis vérifier le résultat obtenu lors de l'accès à l'URL `http://127.0.0.1/webtests/accueil.html` dans un navigateur *web*.

```
<html>
  <head>
    <title>web tests</title>
  </head>
  <body>
    <h1>Tests de l'environnement applicatif</h1>
  </html>
```

26. [devadmin] Vérifier le bon respect des recommandations du W3C (validator.w3.org) et procéder aux corrections si nécessaire.
27. [devadmin] Créer le fichier `infos-env.php`, dont le code est donné ci-dessous, dans le dossier `/var/www/html/webtests`, puis vérifier le résultat obtenu lors de l'accès à l'URL `http://127.0.0.1/webtests/infos-env.php` dans un navigateur *web*.

```
<?php
    echo phpinfo() ;
?>
```

28. Quel est le rôle de la page affichée dans le navigateur ? Quelle est la version de l'interpréteur PHP installé ?
29. [devadmin] Créer le fichier `heure.php`, dont le code est donné ci-dessous, dans le dossier `/var/www/html/webtests`.

```
<html>
  <head>
    <title>web tests</title>
  </head>
  <body>
    Il est <?php echo date( 'H:i' ) ; ?>.
    <hr/>
  </body>
</html>
```

30. [devadmin] Exécuter le script `heure.php` au moyen de la commande `php`, puis observer le résultat obtenu.
31. [devadmin] Cibler l'URL `http://127.0.0.1/webtests/heure.php` dans un navigateur *web*, puis afficher le code source de la page.
32. [devadmin] Créer le fichier `liste-postes.php`, dont le code est donné ci-dessous, dans le dossier `/var/www/html/webtests`.

```
<!DOCTYPE html>

<?php
    $dsn = "mysql:host=localhost;dbname=bdtests" ;
    $connexion = new PDO( $dsn , "superviseur" , "azerty" ) ;
    $select = $connexion -> query("select * from Poste") ;
?>

<html lang="fr">
    <head>
        <meta charset="utf-8">
        <title>XILIM</title>
    </head>
    <body>
        <h1>Postes :</h1>
        <ul>
            <?php
                while( $unPoste = $select -> fetch( PDO::FETCH_OBJ ) ){
            ?>
            <li>
                <?php echo $unPoste -> id . ' ' . $unPoste -> nom ; ?>
            </li>
            <?php } ?>
        </ul>
    </body>
</html>
```

33. [devadmin] Exécuter le script liste-postes.php au moyen de la commande php, puis observer le résultat obtenu.
34. [devadmin] Cibler l'URL `http://127.0.0.1/webtests/liste-postes.php` dans un navigateur *web*, puis afficher le code source de la page.
35. [devadmin] Copier le fichier liste-postes.php dans le fichier enum-postes.php.
36. [devadmin] Modifier le code source du script enum-postes.php de façon à visualiser une liste numérotée des postes recensés dans la base de données.
37. [devadmin] Exécuter le script enum-postes.php au moyen de la commande php, puis observer le résultat obtenu.
38. [devadmin] Cibler l'URL `http://127.0.0.1/webtests/enum-postes.php` dans un navigateur *web*, puis afficher le code source de la page.
39. [devadmin] Modifier le code source du script enum-postes.php de façon à visualiser toutes les informations (identifiant, nom et salle) de chacun des postes recensés dans la base de données.
40. [devadmin] Exécuter le script enum-postes.php au moyen de la commande php, puis observer le résultat obtenu.
41. [devadmin] Cibler l'URL `http://127.0.0.1/webtests/enum-postes.php` dans un navigateur *web*, puis afficher le code source de la page.
42. [devadmin] Copier le fichier enum-postes.php dans le fichier tab-postes.php.
43. [devadmin] Modifier le code source du script tab-postes.php de façon à visualiser toutes les informations de chacun des postes recensés dans la base de données sous forme tabulaire.
44. [devadmin] Exécuter le script tab-postes.php au moyen de la commande php, puis observer le résultat obtenu.
45. [devadmin] Cibler l'URL `http://127.0.0.1/webtests/tab-postes.php` dans un navigateur *web*, puis afficher le code source de la page.